

## Tentamen i Programmering i R, 7.5 hp

---

Skrivtid: 8.00-12.00  
Hjälpmedel: Inget tryckt material, dock finns "R reference card v.2" och några andra referenskort tillgängliga elektroniskt.  
Betygsgränser: Tentamen omfattar totalt 20 poäng. 12 poäng ger Godkänt, 16 poäng ger Väl godkänt.  
Tänk på följande:  
Skriv dina lösningar i **fullständig och läsbar kod**. Kommentera din kod och använd en god kodstil.  
Spara filen med namnet **tentaX.R** där **X** är ditt SC-nummer (klientnummer). Det numret kan du se i studentklienten. Exempel: om du har SC-nummer SC12345 så ska filen heta **tentaSC12345.R**  
Filen lämnas sedan in via studentklienten. Notera att du ska lämna in **en** fil med alla dina lösningar.  
När du har loggat ut från datorn är tentan avslutad. Frågor kan ställas till lärare via studentklienten.  
Spara filer i mappen `/home/student_tilde/`  
**Kommentera direkt i din R-fil** när något behöver förklaras eller diskuteras.  
Eventuella grafer som skapas under tentans gång behöver **INTE** skickas in för rättning, det räcker med att skicka in den kod som producerar figurerna.

---

**OBS: Glöm inte att spara din fil ofta! Om R krashar kan kod förloras.**

### 1. Datastrukturer och beräkningar (4p)

- (a) Räkna ut följande uttryck

$$y = \frac{\sin(x)}{x} + \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

och avrunda till 4 decimaler, där  $x = 1.789$  och där  $e$  är Eulers tal. **1p**

- (b) Skapa en array med dimensionerna (4,5,2) och fyll den med 40 slumpstal från en den uniforma fördelningen  $U(\min = 0, \max = 1)$ . Döp den till `my_array`. **1p**

```
dim(my_array)

[1] 4 5 2
```

- (c) Läs in det inbyggda datamaterialet `diamonds` i R (finns i paktet `ggplot2`). Använd sedan funktioner ur `dplyr` för att göra följande: Välj ut de observationer där variabeln `color` har värdet "E" och variabeln `depth`  $\geq 65$ . Välj sedan ut variablerna `price`, `color` och `depth`. Skapa två nya variabler som du lägger till datamaterialet: `price^2` och `log(depth)`, döp dem till `x1` och `x2` respektive. Sortera efter `price` i fallande ordning. Spara det transformerade datamaterialet som `my_diamonds`. **2p**

```
head(my_diamonds,4)

# A tibble: 4 x 5
  price color depth      x1      x2
  <int> <ord> <dbl>    <dbl> <dbl>
1 15584 E      65.4 242861056  4.18
2 15330 E      65.7 235008900  4.19
3 12829 E      65.0 164583241  4.17
4 12799 E      65.4 163814401  4.18

tail(my_diamonds,4)

# A tibble: 4 x 5
  price color depth      x1      x2
  <int> <ord> <dbl>    <dbl> <dbl>
1   512 E      65.6 262144  4.18
2   492 E      65.0 242064  4.17
3   371 E      66.4 137641  4.20
4   337 E      65.1 113569  4.18
```

## 2. Kontrollstrukturer (4p)

- (a) Skapa en for-loop som gör följande: Utgå från det engelska alfabetet (a,bc,...,x,y,z). I varje iteration ska loopen skriva ut de  $i$  första bokstäverna, där  $i$  är loopindex. Den ska även skriva ut antalet vokaler (a,e,i,o,u,y) varje iteration. Se nedan för hur formen på utskriften ska se ut. Observera att endast de 10 första iterationerna visas nedan. **2p**

```
[1] "Antal vokaler: 1 , a"
[1] "Antal vokaler: 1 , a b"
[1] "Antal vokaler: 1 , a b c"
[1] "Antal vokaler: 1 , a b c d"
[1] "Antal vokaler: 2 , a b c d e"
[1] "Antal vokaler: 2 , a b c d e f"
[1] "Antal vokaler: 2 , a b c d e f g"
[1] "Antal vokaler: 2 , a b c d e f g h"
[1] "Antal vokaler: 3 , a b c d e f g h i"
[1] "Antal vokaler: 3 , a b c d e f g h i j"
```

- (b) Du ska nu skapa en matris  $\mathbf{X}$ , som har 5 rader och 3 kolumner. Elementen defineras som  $x_{i,j} = (1.5 \cdot i - j) \cdot \cos(\pi \cdot j)$ , där  $i$  är radindex och  $j$  är kolumnindex. Uppgiften ska lösas med en nästlad while-loop (alltså en while-loop i en annan while-loop). Om du gjort rätt ska  $\mathbf{X}$  se ut enligt nedan. **2p**

	[,1]	[,2]	[,3]
[1,]	-0.5	-0.5	1.5
[2,]	-2.0	1.0	0.0
[3,]	-3.5	2.5	-1.5
[4,]	-5.0	4.0	-3.0
[5,]	-6.5	5.5	-4.5

### 3. Strängar och datum (4p)

- (a) Läs in paketen `lubridate` och `stringr` i R. Läs i filen "wind\_temp\_data.csv" i R. Spara i en `data.frame` med namnet `wind_temp_data`. Filen innehåller vind- och temperaturdata från New York under delar av året 1973. **0.5p**
- (b) Utgå från filen som lästes in i (a). Använd funktioner ur R för att svara på frågorna nedan. **1.5p**
- Vilken månad var mediantemperaturen som högst?
  - Vilken är högst? Medeltemperaturen för vardagarna eller för helgerna?
  - Hur många hela veckor var det mellan den dagen som det blåste minst och 1973-09-19?
- (c) Nu ska du skapa funktionen `id_detect(x)`. Funktionen ska undersöka om ett specifikt ID-nummer finns i varje element i en textvektor. ID-nummret ska ha följande form: Först tre små bokstäver, följt av bindestreck, och sedan tre siffror. Tillåtna siffror i nummret är 0 till 5. Utgå ifrån att ID-nummret är omgivet av blanksteg (mellanslag), eller ligger i början eller slutet av strängen. Funktionen ska returnera texten "No ID" om det inte finns ett ID-nummer i ett element, annars ska det aktuella ID-nummret returneras. Se exemplen nedan hur funktionen ska fungera. **2 p**

```
text1<-c("abc-123"," xyz-555","agq-345 ","abcd-123","abc-1234","abc--123")
id_detect(x = text1)

[1] "abc-123" "xyz-555" "agq-345" "No ID" "No ID" "No ID"

text2<-c("abse abc-123","hejabcd-123hej","abc abc-123 abc",
"abc abc-123 abc","abcabc-123 abc","abc.123")
id_detect(x = text2)

[1] "abc-123" "No ID" "abc-123" "abc-123" "No ID" "No ID"

text3<-c("abse abc-126","hejabcd-123hej","abc abcd-123 abc","abc abc-1231 abc")
id_detect(x = text3)

[1] "No ID" "No ID" "No ID" "No ID"

text4<-c("aBc-122","abt-103")
id_detect(x = text4)

[1] "No ID" "abt-103"
```

4. Funktioner: Du skapa en ny klass `company`, en konstruktorfunktion och en print-metod för klassen (dvs `print.company()`). Klassen `company` ska vara en lista med elementen `name`, `age`, `startdate` (i given ordning). Dessa element kan vara vektorer, och varje element representerar en anställd. **4p**

- (a) Skapa nu konstruktorfunktionen, som ska heta `company` och ha argumenten `name`, `age` och `startdate`. Funktionen ska kontrollera följande saker: att `name` är en character, att `age > 0` och `startdate` är av Date-klassen. Om något inte är uppfyllt ska valfria felmeddelanden genereras. Obs: det ska vara ett meddelande för varje test. Eftersom argumenten till `company` kan vara vektorer så räcker det att minst ett element i något argument inte uppfyller kraven för ett felmeddelande ska genereras.

```
test1<-company(name = c("Stine","Lo"),age = c(29,38),
startdate = c(ymd("2010-05-02"),ymd("2017-09-10")))
str(test1)

List of 3
 $ name      : chr [1:2] "Stine" "Lo"
 $ age       : num [1:2] 29 38
 $ startdate : Date[1:2], format: "2010-05-02" "2017-09-10"
 - attr(*, "class")= chr "company"

is.list(test1)

[1] TRUE

names(test1)

[1] "name"      "age"      "startdate"
```

- (b) Skapa nu en print-metod för klassen som har argumenten `x` och `compare`. `x` ska vara ett objekt av klassen `company`, `compare` ska vara en Date-variabel. Se exemplen nedan för hur utskriften ska se ut. `compare` ska användas för räkna ut hur länge som varje person har varit anställd (från `startdate` till `compare`). `no_days` och `no_weeks` visar anställningstiden i dagar och veckor respektive.

```
print.company(test1,compare = ymd("2018-02-05"))

  name age no_days no_weeks
1 Stine 29   2836 405.1429
2  Lo  38    148  21.1429

print(test1,compare = ymd("2018-02-05"))

  name age no_days no_weeks
1 Stine 29   2836 405.1429
2  Lo  38    148  21.1429

print(test1,compare = ymd("2017-12-12"))
```

```

  name age no_days no_weeks
1 Stine  29    2781 397.2857
2   Lo  38     93  13.2857

test2<-company(name = c("a","b","c"),age = c(1,2,3),
startdate = c(ymd("2013-01-02"),ymd("2014-02-03"),ymd("2015-03-04")))
print(test2,compare = ymd("2018-02-05"))

  name age no_days no_weeks
1   a   1    1860 265.714
2   b   2    1463 209.000
3   c   3    1069 152.714

print(test2,compare = ymd("2017-01-01"))

  name age no_days no_weeks
1   a   1    1460 208.5714
2   b   2    1063 151.8571
3   c   3     669  95.5714

```

## 5. Statistik, datahantering och grafik (4p)

- (a) Utgå från det förinstallerade datamaterialet **mtcars** och läs in detta dataset. Denna uppgift baseras på extraproblem från laboration 2. **2p**
- Gör om variabeln som beskriver huruvida bilen har automatisk eller manuell växellåda till en faktorvariabel med etiketterna "automatic" och "manual".
  - Skapa en ny kategorisk variabel som delar in bilar efter om de mindre än eller lika med 130 (brutto) hästkrafter, eller fler. Kalla variabeln **hp\_category**, och grupperna för **low** och **high**. Skapa en korstabell mellan dina två nya variabler.
  - Gör sedan ett chitvå-test på tabellen från (ii). Spara teststatistikan i variabeln **test\_stat** och spara p-värdet i variabeln **p\_value**.
- (b) Ladda in datasetet **faithful**. Denna uppgift baseras på extraproblem från laboration 5. Du ska nu göra två scatterplots, med **waiting** på x-axeln och **eruptions** på y-axeln. Huvudrubrik ska vara "Old faithful". Denna uppgift ska lösas med funktioner ur ggplot2. Det verkar finnas två kluster (grupper) i data (titta i plotten): **2p**
- Gör en scatterplot där punkterna för de olika klustren har olika färger
  - Gör en scatterplot där punkterna för de olika klustren har olika punktsymboler men samma färg.

**Kom ihåg:** Skriv alla dina lösningar i en körbar **R-fil**. Spara filen med namnet **tentaX.R** där **X** är ditt SC-nummer (klientnummer) Det numret kan du se i studentklienten. Exempel: om du har SC-nummer SC12345 så ska filen heta **tentaSC12345.R** Lämna sedan in din fil via studentklienten. När du har loggat ut från datorn är tentan avslutad.

*Lycka till!*